Записки научных
семинаров ПОМИ
Том 528, 2023 г.

A. Tiskin

THE CHVÁTAL–SANKOFF PROBLEM AS A PROBLEM OF SYMBOLIC DYNAMICS

Abstract. Given two equally long, uniformly random binary strings, the expected length of their longest common subsequence (LCS) is asymptotically proportional to the strings' length. Finding the proportionality coefficient γ, i.e., the limit of the normalised LCS length for two random binary strings of length $n \to \infty$, is a natural problem, first posed by Chvátal and Sankoff in 1975, and as yet unresolved. This problem has relevance to diverse fields ranging from combinatorics and algorithm analysis to coding theory and computational biology. In a previous paper [47], we used methods of statistical mechanics, as well as some existing results on the combinatorial structure of LCS, to link constant γ to the parameters of a certain stochastic particle process. Here, we complement this analysis by presenting a formulation of the problem in the language of symbolic dynamics and cellular automata, and reporting some preliminary results of a computational experiment aimed at improving the existing numerical estimates for γ. We also point out an error in the previous paper [47], which invalidates some of its claims on the properties of γ.

§1. Introduction

The *longest common subsequence (LCS)* for a pair of strings a, b is the longest string that is a (not necessarily consecutive) subsequence of both a and b. Given a pair of strings as input, the *LCS problem* asks for the length of their LCS (finding the actual characters of the LCS is not required). The LCS problem is a fundamental problem for both theoretical and applied computer science, and for computational molecular biology; it is also a popular programming exercise.

Let strings a, b be of length n, uniformly random over the binary alphabet. Chvátal and Sankoff [14] (see also [53, Chapter 1]) have shown that the expected LCS length of a, b is asymptotically proportional to n. The

Key words and phrases: random strings, longest common subsequence, the Chvátal–Sankoff problem, particle processes, symbolic dynamics, cellular automata.

This work was supported by the Russian Science Foundation under grant no. 22-21-00669, `https://www.rscf.ru/en/project/22-21-00669`.

Chvátal–Sankoff problem asks for the proportionality coefficient γ, i.e., the limit of the normalised expected LCS length $\frac{\mathbb{E}L_n}{n}$ as $n \to \infty$, where the random variable L_n is defined as the LCS length for strings of length n. Alexander [2] has shown that $0 \leqslant \gamma - \frac{\mathbb{E}L_n}{n} \leqslant O\big(\big(\frac{\log n}{n}\big)^{1/2}\big)$.

The Chvátal–Sankoff problem has relevance to diverse fields ranging from combinatorics and algorithm analysis to coding theory (see e.g. Bukh et al. [9]) and computational biology (see e.g. Pevzner and Waterman [37]). For such a natural and simply posed problem, it seems to be surprisingly elusive: neither an exact value nor any closed-form expression for γ are known, and the existing lower and upper numerical bounds on γ are wide apart.

Organisation of this paper. This paper is a follow-up to [47]. For completeness, we restate the related work, main definitions and analysis in Sections 2–7 of this paper. In Sections 8–9, we complement that analysis by presenting a formulation of the problem in the language of symbolic dynamics and cellular automata, and reporting some preliminary results of a computational experiment aimed at improving the existing numerical estimates for γ. In Section 10, we point out an error in [47], which invalidates some of its claims on the properties of γ.

§2. Related work

LCS combinatorics. An important combinatorial feature of the LCS problem, also relevant to its computational aspect, is the problem's close connection with transposition networks and the Hecke monoid (also called the seaweed monoid or the sticky braid monoid). This connection has been explored over decades from different angles and using greatly varying terminology. In the rest of this paper, we will describe this connection in more detail, and will use it as the first step on our path to the Chvátal–Sankoff problem.

While the computational aspect of the LCS problem is outside the scope of this paper, it should be mentioned that the problem's computational complexity, along with that of the closely related edit distance and sequence alignment problems, has been thoroughly studied and is well-understood. Seminal work on LCS algorithms and lower bounds includes e.g. [1, 7, 33, 57].

Random LCS on permutation strings. Apart from binary strings, a question analogous to the Chvátal–Sankoff problem can be asked about pairs of uniformly random permutations of the alphabet $\{1, \ldots, n\}$. The LCS problem on such permutation strings is equivalent to finding the longest increasing subsequence (LIS) of a single permutation of length n. The LCS (respectively, LIS) length in this case turns out to be asymptotically proportional to $\sqrt{n}$. The proportionality constant was found to be exactly 2 in the classical works of Vershik and Kerov [56] and Logan and Shepp [29] (see also [40]), as part of a solution for the more general problem asking for the limit shape of a random Young diagram sampled from the Plancherel distribution.

Bounds and estimates for γ. Chvátal and Sankoff [14] gave the first analysis of the problem, and proved the existence of the limit γ. Properties of the convergence of the normalised LCS length to this limit were studied since then by numerous researchers. Table 1 lists some results on specific lower and upper bounds, as well as experimental numerical estimates of γ.

Reference	$\gamma >$	$\gamma \approx$	$\gamma <$
Chvátal and Sankoff [14]	0.697844	0.8082	0.866595
Deken [19]	0.7615		0.8575
Steele [46] (conjecture attr. to Arratia)	$\gamma \overset{?}{=} 2(\sqrt{2}-1) = 0.8284\ldots$ (*)		
Dančík [17]; Paterson and Dančík [35]	0.77391	0.812	0.83763
Baeza-Yates et al. [4]		0.8118	
Boutet de Monvel [18]		0.812282	
Bundshuh [10]		0.812653	
Lueker [27]	0.788071		0.826280, refutes (*)
Bukh and Cox [8]		0.8122	
this work		0.81169... 0.81175... 0.81182... 0.81187... ... → γ	

Table 1. Bounds and estimates on γ

The best currently known analytic bounds on γ are due to Lueker [27]. Despite the ingenious methods of obtaining these bounds and numerous related results, the gap between the upper and the lower bounds remains quite wide: in particular, not a single digit of γ after decimal point is known exactly.

Stochastic evolution models. Due to the combinatorial properties of the LCS problem that will be presented in the next section, the Chvátal–Sankoff problem turns out to be closely related to the theory of stochastic evolution models, which is a vast and actively developing field of study. Particularly relevant areas within this field include particle processes, random Young diagrams, stochastic cellular automata. Asymptotic properties of such models are studied with the help of partial differential equations (PDEs), which describe a model's evolution at the macroscopic level.

In the rest of this paper, we will describe these connections in more detail, and will build upon them to obtain a restatement of the problem in the language of symbolic dynamics, as well as a new type of computational experiment, aimed at improving the existing numerical estimates for γ.

§3. Combinatorics of the LCS problem

LCS grid. Let strings a, b be of length m, n respectively. The *LCS grid* defined by a, b is a directed graph on an $(m+1) \times (n+1)$ grid of nodes; we visualise the nodes as being indexed top-to-bottom and left-to-right. Every pair of horizontally or vertically adjacent nodes are connected by an edge, directed rightwards (respectively, downwards). A pair of diagonally adjacent nodes (i,j), $(i+1,j+1)$, $0 \leqslant i < m$, $0 \leqslant j < n$, are connected by an edge whenever $a_i = b_j$ (the two characters *match*); this edge is directed towards below-right. The LCS grid can also be viewed as an $m \times n$ grid of cells, each formed by a quadruple of adjacent nodes and their four connecting horizontal and vertical edges. The cell is called *match cell*, if the two corresponding characters match (and therefore the cell contains a diagonal edge), otherwise a *mismatch cell.* The LCS problem is equivalent to asking for the length of a path in the LCS grid from the top-left node $(0,0)$ to the bottom-right node (m,n), that maximises the number of diagonal edges along the path.

Example 1. Figure 1 (left) shows the LCS grid for a pair of binary strings. The horizontal and vertical edges are shown in light-blue, and the diagonal edges in solid red. The left-to-right, top-to-bottom direction of the edges is left implicit.

Sticky braids. The combinatorial structure of the LCS problem is described algebraically by the *Hecke monoid* (also known as the *sticky braid monoid*), which is defined similarly to the classical braid group, but with element inversion replaced by the idempotence relation on the monoid's

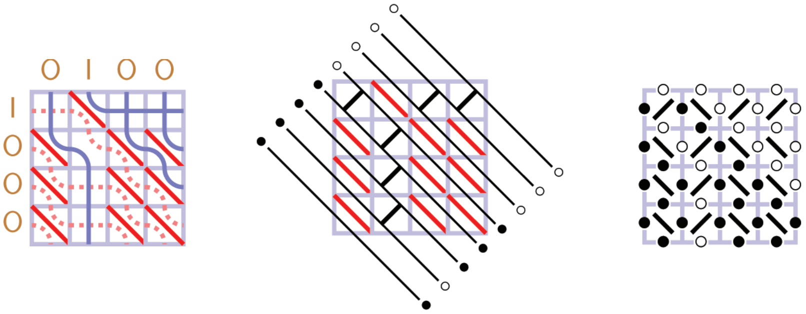


Figure 1. LCS grid with a sticky braid (left), transposition network (centre), particle evolution model (right) for strings $a =$ “1000”, $b =$ “0100”.

generators. Given an LCS grid, strands of the corresponding sticky braid are formed by paths in the dual graph, i.e., the plane graph whose nodes are the faces of the LCS grid, and the edges go across the edges of the LCS grid. Multiplication of sticky braids in the Hecke monoid (also known as *Demazure multiplication*) describes precisely how LCS lengths of input strings and their substrings behave under string concatenation.

Example 2. Figure 1 (left) shows a sticky braid embedded into the LCS grid of the previous example. The braid's strands are shown in darker blue and dotted red.

The connection outlined above between the LCS problem and the Hecke monoid has been rediscovered many times in different forms. In particular, it underlies implicitly the algorithms for various string comparison problems by Schmidt [45], Crochemore et al. [15,16], Alves et al. [3], Hyyrö [23], and was made explicit by Tiskin [48, 51, 54]. More recently, new algorithmic applications of this connection were found by Sakai [42, 43], Tiskin [49, 50, 52, 55], Gawrychowski et al. [21], Hermelin et al. [22], Matarazzo et al. [34], Charalampopoulos et al. [12, 13].

Transposition networks. Another convenient tool for exposing the combinatorial structure of the LCS problem comes in the form of *transposition networks*. These are a special case of comparison networks, which are a classical type of computational circuits studied by Batcher [5], Knuth [24] and many others. In a comparison network, input values travel on an array of parallel wires; any prescribed pair of values can be sorted by a *comparator* connecting their respective wires. In a transposition network, an additional restriction is imposed that only adjacent pairs of wires can be connected by a comparator.

Given a pair of strings a, b of lengths m, n respectively, their LCS grid can be overlayed by a transposition network on $m+n$ wires, extending diagonally from above-left to below-right and passing through the midpoints of the grid's edges. These intersection points of the network's wires and the grid's edges will be called *sites*; we will distinguish horizontal and vertical edge sites. A wire passes through an alternating sequence of horizontal and vertical edge sites; the *value* of a given site is the value carried through it by the wire. A cell is crossed by two wires: one connecting its left and bottom boundary edges, the other its top and right boundary edges. The two sites at the cell's left and top boundary edges are its *entry sites*, and the two sites at its right and bottom boundary edges are its *exit sites*. The network's comparators are specified as follows: a mismatch cell always contains a comparator between the two wires that cross it, while a match cell never contains a comparator. A cell can therefore be of one of two *types*: "match" (denoted '╲'), containing a diagonal grid edge, and "mismatch" (denoted '╱'), containing a network's comparator; the notation indicates the direction of the diagonal edge and of the comparator, respectively. Occasionally, we identify cell type '╲' with value zero, and cell type '╱' with value one.

Example 3. Figure 1 (centre) shows the LCS grid of the previous example, overlaid with its respective transposition network.

Given an input of $m+n$ distinct values sorted in reverse order, the set of values' trajectories through such a transposition network forms a sticky braid corresponding to the comparison of strings a, b; each particular value traces a strand in this braid. The network's output permutation provides detailed information about LCS lengths between various substrings of a, b. For our purposes, the above construction can be simplified as follows: instead of all distinct values, let the transposition network's input consist of m ones, followed by n zeros; note that such an input array is still sorted in reverse. In this context, value zero will be called a *hole* (denoted '$\circ$'), and value one a *particle* (denoted '$\bullet$'). This is done not only to distinguish the (binary) values in the network from (also binary) string characters and (again binary) cell types, but also to reflect in our terminology the important connection with particle interaction models, that we will develop further in the remainder of this paper.

An assignment of values/types to a subset of sites/cells of a transposition network will be called a *configuration*. In particular, the input configuration formed by m particles entering the LCS grid at its left boundary, and n holes entering at the top boundary, will be called the *step initial condition*.

Example 4. The transposition network in Figure 1 (centre) is shown with the step initial condition input sequence at the top-left, and the corresponding output sequence of particles and holes at the bottom-right.

The LCS length of strings a, b is particularly easy to obtain from the transposition network with step initial condition: it is equal to the number of particles among the network's n outputs exiting the grid at the bottom (equivalently, the number of holes among its m outputs exiting the grid at the right). This observation underlies implicitly the bit-parallel LCS algorithms of Crochemore et al. [15] and Hyyrö [23], and was made explicitly e.g. by Majumdar and Nechaev [31] and by Krusche and Tiskin [26]. Let a, b be of equal length $m = n$; in this case, the LCS grid has the shape of a square, and the LCS length is equal to the number of particles (equivalently, the number of holes) that have never crossed the grid's main diagonal.

Example 5. In the previous example, there are three particles among the $n = 4$ outputs at the grid's bottom; the LCS length for strings a, b is also 3. In the course of the evolution of the transposition network, $4 - 3 = 1$ particle has crossed the main diagonal from left to right; accordingly, one hole has done so from top to bottom.

§4. Model *CS*

The combinatorial properties of the LCS problem allow us to reformulate the Chvátal–Sankoff problem in the language of stochastic particle interaction models. By a *network evolution model*, we will understand the evolution of site values from a given input configuration in an infinitely wide transposition network, under a certain probabilistic rule that determines the type of each of the network's cells.

Cell dependencies. Let a, b now be infinite strings, where all characters are independent uniform binary random variables. We define *model CS* (the Chvátal–Sankoff model) as a network evolution model where cell types are determined by character matches and mismatches between strings a, b, as described in the previous section.

Proposition 1. *In model CS, the types of any three distinct cells are mutually independent. The types of any three distinct cells within a ⊞-shape determine uniquely the type of the fourth cell.*

Proof. The first statement is straightforward by the independence and uniformity of character distribution in strings a, b. The second statement is also straightforward, since the sum of the four cells' types must be even. □

In particular, the types of any three cells adjacent in a ⌜-shape are mutually independent; we shall call this property ⌜-*independence*. Note that ⌜-independence relies crucially the uniform distribution of string characters, and would not hold for a non-uniform character distribution, even if it were independent and identical.

Evolution. Let strings a, b be indexed by i, j respectively. The state of model *CS* can be thought of as evolving in several different ways — vertically, horizontally or diagonally, with the discrete time dimension indexed by i, j and $\frac{i+j}{2}$, respectively. We will focus mainly on the diagonal evolution, due to its symmetry and locality properties. The model's state under such evolution corresponds to an anti-diagonal doubly-infinite sequence of particle-hole values, alternating between horizontal and vertical edge sites. Let us index the transposition network's wires entering the grid through its top boundary with nonnegative integers $0, 1, 2, \ldots$, and the wires entering the grid through its left boundary with negative integers $-1, -2, -3, \ldots$; the count in both cases starts from the top-left cell. A time step under diagonal evolution then consists of two half-steps: the first involves comparators operating on pairs of adjacent sites with an odd and an even index (in that order), the second on pairs with an even and an odd index (in that order).

As discussed in the previous section, the behaviour of model *CS* reflects the LCS combinatorics of its underlying string pair a, b.

Proposition 2. *Let $0 \leqslant k \leqslant 2n$. Consider the prefixes of infinite strings a, b of length k, $2n - k$ respectively, and let l be the LCS length of these prefixes. Under diagonal evolution of model CS from step initial condition* *after n time steps, there are $k-l$ particles at sites with indices $2n-2k$ or greater.*

Proof. Well-known from the combinatorial properties of LCS; see e.g. [26, 31]. □

Example 6. Figure 1 (right) shows the evolution of model *CS* from step initial condition on strings a, b of the previous examples. Wires with negative (respectively, nonnegative) indices are those below (respectively, above) the network's main diagonal. Let $n = k = 4$. The LCS length of the input strings, regarded as prefixes of length $k = 2n - k = 4$ of a pair of infinite strings, is $l = 3$; as before, we note that after $n = 4$ time steps, exactly $n - l = 4 - 3 = 1$ particle has crossed over the main diagonal to wires with nonnegative indices.

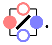
Duality. The definition of model *CS* is symmetric with respect to the reflection of the network about its main diagonal. A pair of configurations will be called *dual*, if one of them is obtained from the other by a reflection about an above-left to below-right axis (exchanging the directions towards below-left and above-right), with simultaneous exchange of sites' values between particles and holes. In particular, the step initial condition is a self-dual configuration.

In the remainder of this paper, we will consider model *CS* with step initial condition. Our analysis will concentrate on the model's behaviour in a small neighbourhood of the main diagonal, where the particle and hole densities should be asymptotically equal by symmetry. Duality will help to simplify the exposition, since in such a setting, a pair of dual configurations will have equal probabilities.

§5. Special notation

Configuration probabilities. We consider configurations of a network evolution model as random events. The probability of an event will be denoted by its graphical representation. Thus, ⧫ = 1 − ○ represents the probability of a given vertical edge site holding a particle, as opposed to a hole, and ⁄ = 1 − ⧵ represents the probability of a given cell being of type "mismatch", as opposed to "match".

We extend this notation to represent conditional probabilities as follows. We juxtapose the conditioning event and the conditioned event in the same picture; the elements of the conditioning event will be highlighted in red, while the elements of the conditioned event will be shown in the ordinary black. For example, the probability of a given cell being of type “mismatch”, conditioned on the cell’s left (respectively top) entry value being a particle (respectively, a hole), will be denoted by $=$ / .

Some events that we consider may be forced by other events: a forced event, conditioned on the forcing event, occurs with certainty. We juxtapose the forcing event and the forced event in the same picture; the elements of the forced event will be highlighted in blue, while the elements of the forcing event will be shown in either black or red, as appropriate. In the previous example, the cell’s exit values are forced: $=$ . Showing forced sub-events is a notational decoration that can formally be omitted; however, it is meant to serve as an intuition aid, especially so when some non-forced sub-event becomes forced in a chain of equalities. For example, we have $=$ $+$.

Annotated equalities. Standard annotated equality $A \overset{\mathsf{def}}{=} B$ (“A is defined as B”) will be used to introduce new notation. Additionally, we will use some other annotations on the equality sign, as an aid to the reader. Notation $A \overset{\mathsf{r}}{=} B$ (“A and B are obtained from each other by reversal with an exchange of particles and holes”) will indicate that the equality holds by the duality property of network configurations.

§6. Scaling limits

Informally, the *scaling limit* of a particle evolution model is the continuous limit of the distribution of particle densities at the model’s sites, as both time and space are simultaneously scaled down at appropriate rates, so that the magnitude of both time and space units tends to zero. A general introduction to the theory of scaling limits is given e.g. by Kriecherbauer and Krug [25].

Scalar conservation laws. Partial differential equations (PDEs) are an indispensable tool in studying the asymptotic behaviour of particle evolution models. Using PDEs, one can relate the global behaviour of the model, such as its non-stationary evolution from a given initial condition, with its local behaviour, such as its stationary state in a small space-time region. A classical example of such a relationship is the asymptotic behaviour of the continuous-time totally asymmetric simple exclusion process (TASEP) with step initial condition, which was shown to be governed by the inviscid Burgers’ equation by Rost [41] (see also [20, 25, 40]).

In general, the scaling limit of a conservative particle model with one spatial dimension can be associated with a *scalar conservation law* (see e.g. [25]), which is a PDE of the form

$$\tfrac{\partial}{\partial t} y + \tfrac{\partial}{\partial x} f(y) = 0$$

where $y = y(t, x)$ is the *density* function of time t and the spatial dimension x, representing the conserved quantity (typically, the mass of some fluid), and $f = f(y)$ is a strictly concave smooth function of density y called the (rightward) *flux*. We are particularly interested in the *step* initial condition:

$$y(0, x) = \begin{cases} 1 & x < 0 \\ 0 & x > 0 \end{cases}$$

In the language of PDEs, the step initial condition is a special case of the Riemann problem for a scalar conservation law. The discontinuity of y at $x = t = 0$ is known as *shock*. This initial shock dissipates over time in a *rarefaction wave*, governed by the equation's solution (see e.g. [25, 44])

$$y(t, x) = \begin{cases} (f')^{-1}(x/t) & f'(1)t \leqslant x \leqslant f'(0)t \\ y(0, x) & \text{otherwise} \end{cases}$$

where f' is the derivative of f, and superscript -1 denotes its functional inverse.

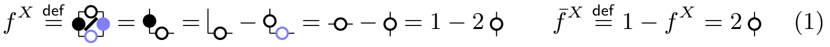
Since the solution scales linearly with t, it is sufficient for the analysis to consider a single time moment $t > 0$; a natural choice is $t = 1$. Let $y(x) = y(1, x)$. We impose further constraints $0 \leqslant y \leqslant 1$, $f(0) = f(1) = 0$, which are natural for the interpretation of y as a fluid's density. The maximum flux $\tilde{f}$ is determined by $f'(y) = 0$, and is therefore attained at density $\tilde{y} = (f')^{-1}(0) = y(0)$; we will call these *peak flux* and *peak density*, respectively.

Recall that under the step initial condition, all the fluid's mass is concentrated in the negative half-line at time $t = 0$. The key characteristic of the system is the amount of mass transported across the origin to the positive half-line by the time $t = 1$, which turns out to be precisely the peak flux:

$$\int\limits_0^{+\infty} y(x)dx = \int\limits_0^{f'(0)} (f')^{-1}(x)dx = \int\limits_0^{\tilde{y}} f'(y)dy = f(\tilde{y}) - f(0) = f(\tilde{y}) = \tilde{f}$$

We will call the function $1 - f = \bar{f}$ and the value $1 - \tilde{f} = \bar{\tilde{f}}$ respectively the *flux complement* function and the *peak flux complement*. A close relationship between the peak flux complement and the constant γ of the Chvátal–Sankoff problem will be exposed in the rest of this section.

Network model limit. For a network evolution model, density y in the above equations is the limiting marginal probability of a site to contain a particle (as opposed to a hole). The flux for a model X is determined as the (unconditional) probability that a particle and a hole are exchanged by a comparator within the cell. This probability, as well as its complement, have a straightforward expression in terms of marginal site probabilities:



$f^X \overset{\text{def}}{=}$
$= 1 - 2$
$\bar{f}^X \overset{\text{def}}{=} 1 - f^X = 2$
(1)

(This form of expression simplifies the one given in [47].) For a model evolving vertically or horizontally, every cell is accounted for in the above expession for the flux in a given time step. For a model evolving diagonally, one half of the cells is accounted for in the first half-step of a time step, and the other half of the cells in the second half-step.

For a model that has mirror symmetry of cell type probabilities about the main diagonal (such as model *CS* and all the others considered in this paper), and that evolves diagonally from the (skew-symmetric) step initial condition, the site probabilities will be skew-symmetric about the main diagonal: particle probability at a site on one side of the main diagonal must be equal to the hole probability at the symmetrically opposite site. By symmetry, the peak density for such a model in the scaling limit is $\tilde{y} = \frac{1}{2}$, realised in a small neighbourhood of the main diagonal.

From now on, we will consider the model's state in an infinitesimally small neighbourhood of the scaling limit point $t = 1$, $x = 0$ on the main diagonal. At that point, both the model's peak flux and peak density are realised, so we will write simply y for $\tilde{y}$ and f^X for $\tilde{f}^X$. The peak density y is composed from particle probabilities at horizontal and vertical sites, or, symmetrically, particle and hole probabilities at just the horizontal, or just the vertical sites: $y = u + \bar{u} =$ -●- $+$ ●⃒ $=$ -●- $+$ -○- $=$ ϕ $+$ ●⃒ $= \frac{1}{2}$. The evolution of the model in such a small neighbourhood can be considered to be in a stationary state; we will use this stationarity to derive the joint distribution for site probabilities of our models.

A limit for model ***CS*****.** In general, finding an explicit flux function for a particle evolution model may be difficult, and even the convergence to a scaling limit is not guaranteed. Fortunately, the existence of a continuous scaling limit for model CS follows directly from Proposition 2. Indeed, the model's convergence at a point on the main diagonal is equivalent to the convergence of scaled LCS length for a pair of equally long uniformly random binary strings, i.e., to the existence of constant γ. As mentioned in the Introduction, this was established already by Chvátal and Sankoff [14] (see also [53, Chapter 1]). In much the same way, the model's convergence at any other point is equivalent to the convergence of scaled LCS length for a pair of random binary strings with a given limiting ratio of their lengths, which can be established by a slight modification of the same proof.

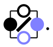
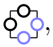
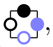
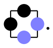
The Chvátal–Sankoff problem can now be reformulated as finding the peak flux complement $\gamma = \bar{f}^{CS}$ for model CS.

§7. Model B

In keeping with the traditional terminology, let us define *model B* (the Bernoulli model) as the network evolution model, where a cell is assigned type "mismatch" with a fixed probability $p \overset{\mathsf{def}}{=} \diagup$, called the model's *(jump) rate*, independently of any site values or types of any other cells (this initial definition will be generalised later). Intuitively, every cell tosses an independent biased coin p to determine its type.

Model B has been applied to the study of the Chvátal–Sankoff problem by Boutet de Monvel [18], Majumdar and Nechaev [31], Priezzhev and Schütz [38], Bukh and Cox [8]. It is closely related to a classical particle model known as the *totally asymmetric simple exclusion process (TASEP)*. The TASEP consists of an of array of sites, each occupied by a particles or a hole. It evolves by a particle jumping at a random time into a hole on its right; symmetrically, the hole "jumps" to its left to the site previously occupied by the particle. Updates may occur in continuous time (classical TASEP, which we do not consider any further) or in discrete time (DT-TASEP). Within a time step of DT-TASEP, the update policy may be parallel (the process also known as multi-corner growth of a Young diagram, which we do not consider any further), forward-sequential, backward-sequential, or sublattice-parallel. The latter three update policies essentially only differ by a change of coordinates, and correspond to model B evolving vertically, horizontally or diagonally, respectively. An analysis of DT-TASEP with different update policies has been given by Rajewsky et al. [39] and by Martin and Schmidt [32]. Model B and DT-TASEP can be considered as a special case of the six-vertex model analysed by Borodin et al. [6], with weights assigned according to measure $\mathcal{P}(p, 0)$ defined therein.

Model B and other network evolution models presented in this paper can also be considered as special cases of stochastic cellular automata (see e.g. [11, 30]). However, the simplifying "well-mixing" assumptions, that are usually made in that context, do not hold for our models.

Cell type probabilities. We note that a cell's type only affects the model's behaviour when its entry pair is , distinguishing the events and . For any other entry pairs, the cell's exit values are forced by the entry values and are independent of the cell's type: the corresponding events are , , . In these cases, the cell's type probability can be set differently from p, without affecting the model's behaviour. Therefore, we can generalise the definition of model B by introducing a formal dependency of a cell's type on its entry pair, while making sure that the model's new definition is still invariant with respect to duality of configurations.

Definition 1. We say that a cell's type *depends exclusively* on a set of sites' values in a given half-step, if, conditioned on this set, it is conditionally independent of any other site values in the same half-step.

We define $4 = 1 \cdot 2 + 2$ (one dual pair and two self-dual singletons) conditional probabilities for a cell's type, specifying its exclusive dependence on the entry site pair:

$p_0 \overset{\mathsf{def}}{=}$ $\overset{\mathsf{r}}{=} p_3 \overset{\mathsf{def}}{=}$ $p_1 \overset{\mathsf{def}}{=}$ $p_2 \overset{\mathsf{def}}{=}$

The subscripts correspond to the entry pair values being read as a two-digit binary number, bottom-left to top-right: $p_0 = p_{\circ\circ}$, etc. Intuitively, a cell now has four biased coins p_0, p_1, p_2, p_3, including a dual pair $p_0 \overset{\mathsf{r}}{=} p_3$. The cell reads its entry pair (as a binary number), and then tosses the corresponding coin to determine its type; the combination of the cell's entry pair and its chosen type then determines the cell's exit pair.

Conditional probability p_2 corresponds to the rate p in the original definition of model B, and determines solely the model's behaviour (in particular, its flux). We will therefore reserve the term *rate* for p_2, whereas the remaining conditional probabilities $p_0 \overset{\mathsf{r}}{=} p_3$, p_1 will be called *pseudo-rates*. These pseudo-rates do not affect the behaviour of the model, and therefore can temporarily be left unconstrained. This leaves us the freedom to set them later, in an attempt to fit model B to the constraints of model CS. (This idea has been explored in [47, Section 8], and is not necessary for this paper; however, we keep the notation for consistency, and for future reference).

Alternating sequences. Our models, including model B, will have time-invariant distributions satisfying the following natural property.

Definition 2. An *alternating sequence* is a doubly-infinite sequence of (generally dependent) particle-hole random variables (ξ_i), $i \in \mathbb{Z}$, that is invariant with respect to

- a shift by 1, mapping $i \mapsto i + 1$
- a reversal about $\frac{1}{2}$, mapping $i \mapsto -i + 1$

both of these with simultaneous exchange between particles and holes.

Note that both a shift and a reversal of the given type flip the parity of indices. Definition 2 implies that an alternating sequence is also invariant with respect to arbitrary shifts and reversals, where holes and particles are exchanged if and only if the parity of indices is flipped.

We consider alternating sequences of site values in a given half-step of diagonal network evolution, identifying arbitrarily the even (respectively, odd) indices of the sequence with the horizontal (respectively, vertical) edge sites. Annotated equality $A \overset{\mathsf{r}}{=} B$, when applied to such sequences, will stand for "A and B are obtained from each other by a parity-exchanging reversal with a simultaneous exchange between particles and holes"; this is consistent with the previous usage of this notation to express duality of configurations. Furthermore, annotated equality $A \overset{\mathsf{s}}{=} B$ will have similar meaning, but with a reversal replaced by a shift. Notation $A \overset{\mathsf{sr}}{=} B$ will be used when $A \overset{\mathsf{s}}{=} B$ and $A \overset{\mathsf{r}}{=} B$ are both applicable.

We denote the marginal site probabilities by

$$u \overset{\mathsf{def}}{=} \overset{|}{\underset{|}{\circ}} \overset{\mathsf{sr}}{=} -\!\bullet\!- \qquad \bar{u} \overset{\mathsf{def}}{=} \overset{|}{\underset{|}{\bullet}} \overset{\mathsf{sr}}{=} -\!\circ\!- \tag{2}$$

Substituting (2) into (1), we obtain a simple expression for the peak flux complement of our models.

Proposition 3. *Let X be a network evolution model in a stationary state, where the time-invariant distribution of site values is given by an alternating sequence. Then the peak flux complement is*

$$\bar{f}^X = -\!\bullet\!- + \overset{|}{\underset{|}{\circ}} = u + u = 2u$$

AB sequences. We first consider the most basic special case of an alternating sequence.

Definition 3. An alternating sequence (ξ_i), $i \in \mathbb{Z}$, is an *AB (alternating Bernoulli) sequence*, if all its elements are mutually independent.

In particular, an AB sequence of site values in a given half-step of a network evolution model is a product measure with marginal site probabilities (2).

Time invariance. Consider the evolution of model B on an AB sequence in a stationary state. The model's rate and the site densities of the sequence are connected by the *time-invariance equation*:

$uu =$ $=$ $= \bar{u}\bar{u}\bar{p}_2$ (3)

We recall a well-known result on the time-invariant distribution for the diagonal evolution of model B (see e.g. Rajewsky et al. [39], Martin and Schmidt [32]).

Theorem 1. *An AB sequence with parameter u determined by* (3) *is a time-invariant distribution for model B with a given rate p_2.*

Proof. It is sufficient to show that the AB property is preserved in a single half-step of the evolution of model B. The independence between site values a, b at the end of the half-step in a configuration ${}^{\ulcorner}_{a}{}^{b-}$ is obvious, since these values are obtained in different cells; independence in a configuration ${}_{-a}{}^{\dot{b}}$ is established by (3). In the equation (3), the left-hand side expresses the AB property in a configuration at the half-step's end, while the right-hand side relies on the same property at the half-step's beginning. □

The Arratia–Steele conjecture. Since the marginal probabilities of cell types in model CS are all equal to $\frac{1}{2}$, and since these types are -independent and, more generally, three-wise independent, at some point it was quite natural to conjecture that all cell dependence (i.e., ⊞-dependence and, more generally, all four-wise and higher-order dependence) could also be ignored, so that model CS would be equivalent to model B with $p = p_2 = \frac{1}{2}$, which we denote $B(1/2)$. Substituting $p_2 = \frac{1}{2}$ into (3), we obtain a quadratic equation that gives us the peak site marginal probability and the peak flux complement for model $B(1/2)$, which can be considered as an approximation for γ:

$$u = \sqrt{2} - 1 = 0.414213\ldots \qquad \gamma \approx \bar{f}^{B(1/2)} = 2u = 2(\sqrt{2} - 1) = 0.828427\ldots$$

The conjecture, attributed to Arratia by Steele [46], was that the above expression gives the exact value of γ. This conjecture was disproved by the upper bound $\gamma \leqslant 0.826280$ due to Lueker [27].

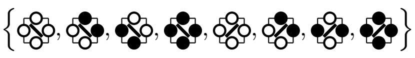
§8. Models B and CS as measures on shift spaces

In this section, we rely on the textbook by Lind and Marcus [28] for the definitions and background.

Let A be a finite *alphabet*. The *full one-dimensional (1D) shift* $A^{\mathbb{Z}}$ is the set of doubly-infinite sequences of symbols from A, equipped with the prodiscrete topology. A *1D shift* (also *subshift*, short for *shift space*) is a closed subset of $A^{\mathbb{Z}}$ that is invariant under the action of the shift operator $x \mapsto x+1$. A *2D shift* is defined similarly as a closed subset of the *full two-dimensional (2D) shift* $A^{\mathbb{Z}^2}$, invariant under the action of the shift operators $(x,y) \mapsto (x{+}1, y)$ and $(x,y) \mapsto (x, y{+}1)$. A 1D (respectively, 2D) shift X can be specified by a set F of *forbidden* finite words (respectively, patterns). A shift is a *shift of finite type (SFT)*, if such a set F can be chosen to be finite.

Given a shift, we are interested in shift-invariant probability measures that can be defined on it. Among these, of particular interest are *measures of maximum entropy (MME)*. It is well-known that MME exist for both 1D and 2D SFTs. However, there is a substantial difference in their level of complexity. For a 1D SFT, an MME corresponds to a Markov chain of finite order, and can be obtaned from its set of forbidden words explicitly via the Perron–Frobenius theorem. For a 2D SFT, no explicit constructions are known except for a few specific cases; many aspects of 2D SFTs are algorithmically undecidable.

2D shifts. Both models that we consider in this paper can naturally be defined on 2D shifts. The alphabet will consist of eight symbols, each representing a cell's type and its boundary sites' contents:

$A =$ 

Model B is specified by the set F_B that forbids any □□- or ⊟-shape of cells that do not agree on the contents of their shared site. Model CS is specified by the set F_{CS} that forbids, in addition, also any ⊞-shape that contains an odd number of cells of each type ('$\diagdown$' and '$\diagup$'):

$$F_{CS} = F_B \cup \left\{ \begin{array}{c|c} \diagdown & \diagdown \\ \hline \diagdown & \diagup \end{array}, \begin{array}{c|c} \diagdown & \diagup \\ \hline \diagdown & \diagdown \end{array}, \begin{array}{c|c} \diagdown & \diagdown \\ \hline \diagup & \diagdown \end{array}, \begin{array}{c|c} \diagup & \diagdown \\ \hline \diagdown & \diagdown \end{array}, \begin{array}{c|c} \diagup & \diagup \\ \hline \diagup & \diagdown \end{array}, \begin{array}{c|c} \diagup & \diagdown \\ \hline \diagup & \diagup \end{array}, \begin{array}{c|c} \diagup & \diagup \\ \hline \diagdown & \diagup \end{array}, \begin{array}{c|c} \diagup & \diagup \\ \hline \diagup & \diagdown \end{array} \right\}$$

Each ⊞-shape in this expression represents a set of patterns that are consistent with it. Clearly, both sets F_B and F_{CS} are finite, therefore the shifts they specify are both SFTs.

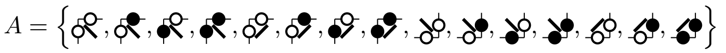
Stationary distributions in models B and CS correspond to certain invariant probability measures on their 2D shifts. A necessary property that such a measure must satisfy, apart from shift-invariance, is that it must be uniform when restricted to the cells' types, "erasing" the sites' contents. Formally, we define a canonical map from alphabet A to a two-symbol alphabet:

$$f : A \to \{\diagdown, \diagup\}$$

This map lifts to a factor map between 2D shifts, which we also denote f. The stationary distributions in models B and CS in a small neighbourhood of the main diagonal can now be characterised (see e.g. [36]) as a shift-invariant measure μ of maximal relative entropy (MMRE) on their respective shifts, subject to the pushforward measure of μ under f being a uniform measure.

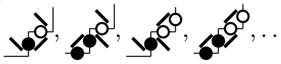
1D shifts. It is also quite natural to define both models on 1D shifts. In general, such shifts are conceptually simpler than 2D ones; however, the complexity of the models' definitions increases relative to the 2D case. In contrast to the 2D case, where a shift's single element defines the whole evolution of the model, here we consider the state of the model's evolution at a given time step. Therefore, a shift's element will correspond to an antidiagonal configuration of cells. Space invariance of the model will correspond to the invariance under the 1D shift operator, while the time invariance will have to be captured separately.

The alphabet will now consist of 15 symbols:



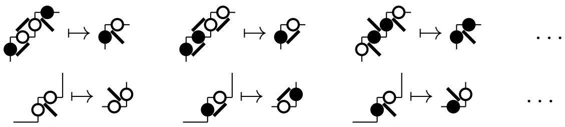
These symbols will be joined up to form an infinite zigzag antidiagonal strip in the grid. The sub-alphabet of the first eight symbols represents the (say) odd positions of the strip, and one of the remaining seven symbols the (complementary) even positions; the parity of the positions will change under the shift operator, but the alternation of the sub-alphabets will not.

Model B is specified by the set F_B that forbids any □□- or ⊟-shape of cells that do not respect the alternation of the sub-alphabets or do not agree on the contents of their shared site. Model CS is specified by the set F_{CS} that forbids, in addition, also an infinite set of words that are inconsistent with the current configuration resulting from the model's evolution over an infinite number of steps back in time:

$F_{CS} = F_B \cup \Big\{$  $\Big\}$

Set F_B is finite, and therefore its corresponding shift is an SFT. However, F_{CS} is infinite, and clearly cannot be reduced to any finite set specifying the same shift. Thus, its corresponding shift is not an SFT.

In order to capture the models' time invariance, we define a *cellular automaton* map $T : A^3 \to A$, which describes the models' evolution over a time step in an obvious way. Some examples of argument-value pairs under this map are



We say that a measure μ is *T-invariant*, if $\mu(u) = \sum_{v \in T^{-1}(u)} \mu(v)$, whenever the left-hand side (and therefore also the right-hand side) is defined.

Finally, as with the 2D version of the shifts, we define a canonical map from alphabet A to a two-symbol alphabet:

$$f : A \to \{\diagdown, \diagup\}$$

This map lifts to a factor map between 1D shifts, which we also denote f. The stationary distributions in models B and CS in a small neighbourhood of the main diagonal can now be characterised as a shift-invariant, T-invariant measure μ of maximal relative entropy (MMRE) on their respective shifts, subject to the pushforward measure of μ under f being a uniform measure.

§9. Numerical experiments

In the previous section, we have characterised the stationary distrubution in model CS as a measure with certain invariance and relativisation properties on both a 2D and 1D shift. While the characterisation via a 2D shift is conceptually simpler, 2D shifts themselves are in general much less amenable to analysis than 1D ones. In particular, this characterisation does not appear to provide a good way to approximate the required measure numerically, even though it is defined on an SFT.

In contrast, the characterisation of model CS via a 2D shift requires an additional property of T-invariance, and its underlying shift is not an SFT. However, having a clearly defined set of forbidden words F for this shift suggest a natural technique for approximating the model's stationary distribution by obtaining measures with the required properties on a decreasing sequence of 1D SFTs, beginning with the shift underlying model B, and each next shift in the sequence being defined by a finite subset of F, consisting of words no longer than a specified length $n \to \infty$. Within a certain range of values of n, a stationary distribution of the corresponding evolution model approximating model CS can be obtained by iterating the action of the cellular automaton map T. This gives us a reasonably fast convergence to the parameters Markov chain providing the stationary distribution, allowing us to approximate it either to the machine precision, or, for somewhat higher values of n, to at least five decimal points.

The described technique has been implemented as a program in C++, incorporating a number of quite sophisticated time and memory optimisations. Table 1 gives a sequence of approximations to γ that we have obtained with our implementation for $n = 10, 12, 14, 16$. Each number gives an approximation to γ obtained from the corresponding SFT, accurate to all the decimal points provided. The time and memory required by the computation grows exponentially; the last approximation in the sequence required several hours of execution parallelised over an 8-core processor of an Intel Xeon Gold server. The convergence of the sequence to the value of γ is relatively slow; however, by scaling up our experiment to a high-performance computing server, we hope to obtain an approximation for γ accurate to at least four decimal points.

§10. Model M: a correction

In [47], we introduced a further model that we called model M. It is essentially a Markov-chain model of the type described in the previous section, that approximates model CS and corresponds to an SFT defined by a finite subset of the set of forbidden words for model CS. Unfortunately, it was claimed in [47] erroneously that the two models were equivalent; in fact, this is not the case, and cannot be the case for any SFT-based model. The actual error was in failing to recognise fully the non-local nature of dependencies between the cells in model CS: an attempt to “localise” the dependencies in [47, Section 10], and in particular in the proof of Theorem 3 there, was misguided. The statement of Theorem 3 and its restatements elsewhere in the paper are false.

§11. Conclusion

In this paper, we have restated the Chvátal–Sankoff problem in the language of symbolic dynamics in 1D and 2D, using the problem's connection with stochastic particle processes that we explored previously in [47]. In doing so, we relied on existing results on the combinatorial structure of the LCS problem and the theory of continuous scaling limits for discrete particle processes. Obtaining exact solutions for 2D symbolic dynamics models is notoriously difficult, so such a direct restatement may serve as a partial explanation of the apparent difficulty of the Chvátal–Sankoff problem itself.

We have demonstrated that our restatement of the Chvátal–Sankoff problem as a 1D symbolic dynamics problem lends itself to a new approach to numerical approximation of constant γ. We have reported preliminary results of a numerical experiment based on this approach. We note that our estimate is obtained by a new method, completely different from Monte-Carlo simulation methods that have mostly been employed so far. We also note that the published details of any previous numerical experiments are scarce. Improving the computation efficiency and the convergence properties of our experiment remains a question for further study.

Further challenges outlined in the conclusion of [47] still stand. In particular, it would be interesting to extend our approach to (in increasing order of apparent difficulty)

- strings of unequal lengths;
- Levenshtein distance between strings;
- non-uniform and non-independent character distributions within the strings;
- strings over larger alphabets;
- comparing more than two strings.

We leave these questions open for future work.

References

1. A. Abboud, A. Backurs, V. Vassilevska Williams, *Tight hardness results for LCS and other sequence similarity measures.* — In: Proceedings of FOCS (2015), pp. 59–78.
2. K. S. Alexander, *The rate of convergence of the mean length of the longest common subsequence.* — Annal. Appl. Probab. **4**, No. 4 (1994), 1074–1082.
3. C. E. R. Alves, E. N. Cáceres, S. W. Song, *An all-substrings common subsequence algorithm.* — Discrete Appl. Math. **156**, No. 7 (2008), 1025–1035.

4. R. A. Baeza-Yates, R. Gavaldà, G. Navarro, R. Scheihing, *Bounding the expected length of longest common subsequences and forests.* — Theory of Computing Systems **32**, No. 4 (1999), 435–452.
5. K. E. Batcher, *Sorting networks and their applications.* — In: Proceedings of AFIPS, Vol. 32 (1968), pp. 307–314.
6. A. Borodin, I. Corwin, V. Gorin, *Stochastic six-vertex model.* — Duke Math. J. **165**, No. 3 (2016), 563–624.
7. K. Bringmann, M. Künnemann, *Multivariate fine-grained complexity of longest common subsequence.* — In: Proceedings of ACM-SIAM SODA (2018), pp. 1216–1235.
8. B. Bukh, C. Cox, *Periodic words, common subsequences and frogs.* — Annals Appl. Probab. **32**, No. 2 (2022), 1295–1332.
9. B. Bukh, V. Guruswami, J. Hastad, *An improved bound on the fraction of correctable deletions.* — IEEE Transactions on Information Theory **63**, No. 1 (2017), 93–103.
10. R. Bundschuh, *High precision simulations of the longest common subsequence problem.* — European Phys. J. B **22**, No. 4 (2001), 533–541.
11. J. Casse, *Probabilistic cellular automata with general alphabets possessing a Markov chain as an invariant distribution.* — Adv. Appl. Probab. **48**, No. 2 (2016), 369–391.
12. P. Charalampopoulos, P. Gawrychowski, S. Mozes, O. Weimann, *An Almost Optimal Edit Distance Oracle.* — In: Proceedings of ICALP, Vol. 198 of *Leibniz International Proceedings in Informatics*, pages 48:1–48:20, 2021.
13. P. Charalampopoulos, T. Kociumaka, Sh. Mozes, *Dynamic string alignment.* — In: Proceedings of CPM, Vol. 161, *LIPIcs* (2020), pp. 9:1–9:13.
14. V. Chvátal, D. Sankoff, *Longest common subsequences of two random sequences.* — J. Appl. Probab. **12**, No. 2 (1975), 306–315.
15. M. Crochemore, C. S. Iliopoulos, Y. J. Pinzon, J. F. Reid, *A fast and practical bit-vector algorithm for the Longest Common Subsequence problem.* — Inform. Proc. Letters **80** (2001), 279–285.
16. M. Crochemore, G. M. Landau, M. Ziv-Ukelson, *A subquadratic sequence alignment algorithm for unrestricted score matrices.* — SIAM J. Comput. **32** (2003), 1654–1673.
17. V. Dančík, *Expected Length of Longest Common Subsequences.* PhD thesis, University of Warwick (1994).
18. J. Boutet De Monvel, *Extensive simulations for longest common subsequences: Finite size scaling, a cavity solution, and configuration space properties.* — Europ. Phys. J. B **7**, No. 2 (1999), 293–308.
19. J. G. Deken, *Some limit results for longest common subsequences.* — Discrete Math. **26**, No. 1 (1979), 17–31.
20. Pablo A. Ferrari, *TASEP hydrodynamics using microscopic characteristics.* — Probab. Surveys **15** (2018), 1–27.
21. P. Gawrychowski, *Faster algorithm for computing the edit distance between SLP-compressed strings.* — In: Proceedings of SPIRE, Vol. 7608 of *Lecture Notes in Computer Science*, pages 229–236, 2012.

22. D. Hermelin, G. M. Landau, S. Landau, O. Weimann, *Unified Compression-Based Acceleration of Edit-Distance Computation.* — Algorithmica **65**, No. 2 (2013), 339–353.
23. H. Hyyrö, *Mining bit-parallel LCS-length algorithms.* — In: Proceedings of SPIRE, Vol. 10508, *Lecture Notes in Computer Science* (2017), pp. 214–220.
24. D. E. Knuth, *The Art of Computer Programming: Sorting and Searching*, Vol. 3. Addison Wesley (1998).
25. T. Kriecherbauer, J. Krug, *A pedestrian's view on interacting particle systems, KPZ universality and random matrices.* — J. Phys. A: Math. Theor. **43**, No. 40 (2010), 403001.
26. P. Krusche, A. Tiskin, *String comparison by transposition networks.* — In: London Algorithmics 2008 Theory and Practice, Vol. 11 of *Texts in Algorithmics.* College Publications, 2009.
27. G. S. Lueker, *Improved bounds on the average length of longest common subsequences.* — J. ACM **56**, No. 3 (2009), 17:1–17:38.
28. D. Lind, B. Marcus, *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, second edition (2021).
29. B. F. Logan, L. A. Shepp, *A variational problem for random Young tableaux.* — Adv. Math. **26**, No. 2 (1977), 206–222.
30. J. Mairesse, I. Marcovici, *Around probabilistic cellular automata. Theor. Computer Sci.* **559** (2014), 42–72.
31. S. N. Majumdar, S. Nechaev, *Exact asymptotic results for the Bernoulli matching model of sequence alignment.* — Physical Review E: Statistical, Nonlinear, and Soft Matter Physics **72**, No. 2 (2005), :020901.
32. J. Martin, P. Schmidt, *Multi-type TASEP in discrete time.* — Latin Amer. J. Probab. Math. Statist. **8** (2011), 303–333.
33. W. J. Masek, M. S. Paterson, *A faster algorithm computing string edit distances.* — J. Comput. System Sci. **20**, No. 1 (1980), 18–31.
34. U. Matarazzo, D. Tsur, M. Ziv-Ukelson, *Efficient all path score computations on grid graphs.* — Theor Comput. Sci. **525** (2014), 138–149.
35. M. Paterson, V. Dančík, *Longest common subsequences.* — In: Proceedings of MFCS, Vol. 841 of *Lecture Notes in Computer Science* (1994), pp. 127–142.
36. K Petersen, A Quas, S Shin, *Measures of maximal relative entropy.* — Ergodic Theory and Dynamical Systems **23**, No. 1 (2003).
37. P. A. Pevzner, M. S. Waterman, *Open combinatorial problems in computational molecular biology.* — In: Proceedings of ISTCS (1995), pp. 158–173.
38. V. B. Priezzhev, G. M. Schütz, *Exact solution of the Bernoulli matching model of sequence alignment.* — J. Statist. Mech.: Theory and Experiment **09** (2008), P09007.
39. N. Rajewsky, L. Santen, A. Schadschneider, M. Schreckenberg, *The asymmetric exclusion process: comparison of update procedures.* — J. Statist. Phys. **92** (1998), 151–194.
40. D. Romik, *The Surprising Mathematics of Longest Increasing Subsequences.* Cambridge University Press, Cambridge (2014).

41. H. Rost, *Non-equilibrium behaviour of a many particle process: Density profile and local equilibria.* — Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete **58**, No. 1 (1981), 41–53.
42. Y. Sakai, *A substring–substring LCS data structure.* — Theor. Comput. Sci. **753**, No. 2 (2019), 16–34.
43. Y. Sakai, *A fast algorithm for multiplying min-sum permutations.* Discrete Appl. Math. **159** (2011), 2175–2183.
44. S. Salsa *Partial Differential Equations in Action*, Vol. 99 *UNITEXT*. Springer International Publishing (2016).
45. J. P. Schmidt, *All highest scoring paths in weighted grid graphs and their application to finding all approximate repeats in strings.* — SIAM J. Comput. **27**, No. 4 (1998), 972–992.
46. J. M. Steele, *An Efron-Stein inequality for nonsymmetric statistics.* — Annals Statist. **14**, No. 2 (1986), 753–758.
47. A. Tiskin, The Chvátal–Sankoff problem: Understanding random string comparison through stochastic processes. — *Zap. Nauchn. Semin. POMI* **517** (2022), 191–224.
48. A. Tiskin, *Semi-local longest common subsequences in subquadratic time.* J. Disc. Algorithms **6**, No. 4 (2008), 570–581.
49. A. Tiskin, *Periodic String Comparison.* In: Proceedings of CPM, Vol. 5577, *Lecture Notes in Computer Science* (2009), pp. 193–206.
50. A. Tiskin, *Towards Approximate Matching in Compressed Strings: Local Subsequence Recognition.* — In: Proceedings of CSR, Vol. 6651, *Lecture Notes in Computer Science* (2011), pp. 401–414.
51. A. Tiskin, *Fast Distance Multiplication of Unit-Monge Matrices.* — Algorithmica **71** (2015), 859–888.
52. A. Tiskin, *Bounded-Length Smith–Waterman Alignment.* — In Proceedings of WABI, Vol. 143, *Leibniz International Proceedings in Informatics* (2019), pp. 16:1–16:12.
53. J. M. Steele, *Probability Theory and Combinatorial Optimization*, Vol. 69 of *CBMS-NSF regional conference series in applied mathematics.* SIAM, 1997.
54. A. Tiskin, *Semi-local string comparison: Algorithmic techniques and applications.* — Math. Comput. Sci. **1**, No. 4 (2008), 571–603.
55. A. Tiskin, *Communication vs synchronisation in parallel string comparison.* — In: Proceedings of SPAA, pages 479–489, 2020.
56. A M Vershik, S V Kerov, *Asymptotics of the Plancherel measure of the symmetric group and the limiting form of Young tableaux.* — Dokl. Akad. Nauk **233**, No. 6 (1977), 1024–1027.
57. R. A. Wagner, M. J. Fischer, *The string-to-string correction problem.* — J. ACM **21**, No. 1 (1974), 168–173.

Department of Mathematics
and Computer Science
St. Petersburg State University;
St.Petersburg Electrotechnical University “LETI”
E-mail: alextiskin@gmail.com

Поступило 1 декабря 2023 г.